Shaping of Multidimensional Signal Constellations Using a Lookup Table

A. K. Khandani and P. Kabal

Abstract—**This paper describes a lookup table for the addressing of an optimally shaped constellation. The method is based on partitioning the subconstellations into shaping macro-shells of integer bit rate and increasing average energy. The macro-shells do not need to have an equal number of points. A lookup table is used to select a subset of the partitions in the cartesian product space. By devising appropriate partitioning / merging rules, we obtain suboptimum schemes of very low addressing complexity and small performance degradation. The performance is computed using the weight distribution of an optimally shaped constellation.**

Index Terms—**Lookup table, integer bit rate, nonuniform merging, prefix code.**

I. Introduction

Consider the problem of transmitting the output of a source composed of M equiprobable symbols over a channel. The channel provides us with a given number of dimensions, say N, per signaling interval. For instance in quadrature modulated systems, a block of $N/2$ symbols forms an N-D (N-dimensional) space. To achieve the transmission, we select M points over the channel space. Each of the source symbols is represented by one of these points. This collection of points is called a signal constellation.

In shaping, one tries to reduce the average energy of a signal constellation for a given number of points from a given packing. The price to be paid for the reduction in the average energy (measured by the shaping gain, γ_s) involves: i) an increase in the factor CER_s,[1] (constellation-expansion-ratio), and ii) an increase in the addressing complexity.[2,3] Addressing is often the most difficult task associated with the shaping of a high-dimensional constellation. For example, for 2-D subconstellations composed of 256 points in a 32-D space, a direct addressing scheme using a lookup table requires a block of memory with about 2^{128} memory locations (with each location having a word length of 128 b). In the present work, we introduce suboptimum methods to reduce this memory size to about 0.8 k bytes per 32-D while the degradation in performance is negligible.

A. Previous Work

Conway and Sloane in [2] introduced the idea of Voronoi constellations based on using the Voronoi region of a lattice Λ_s as the shaping region. In the work of Wei [3] shaping is a side effect of the method employed to transmit a nonintegral number of bits per two dimensions. The addressing of this method is achieved by a lookup table. Forney and Wei generalize this method in [1]. Voronoi constellations are further considered by Forney in [4]. In [5], Calderbank and Ozarow introduce a shaping method that is directly achieved on the 2-D subconstellations. In this method, the 2-D subconstellations are partitioned into equal sized subregions of increasing average energy. A shaping code is then used to specify the sequence of the subregions. The shaping code is designed so that the lower energy subregions are used more frequently. The idea of trellis shaping is introduced in [6]. This idea is based on using an infinite-dimensional Voronoi region determined by a convolutional code to shape the constellation. Lang and Longstaff in [7] use an addressing scheme that is based on decomposing the space into lower dimensional subspaces via generating function techniques.

In [8], Kschischang and Pasupathy discuss a shaping method that is based on using the 2-D points with nonequal probability. In [9], Livingston discusses a shaping method in which the 2-D subspaces are partitioned into circular shells of increasing size. In this method, the 2-D shells are used with equal probability inducing a nonuniform distribution on the 2-D points. In a continuation to [5] and [9], Calderbank and Klimesh in [10] use a balanced binary code to select the sequence of the 2-D circular shells. This scheme results in a fixed rate per signaling interval.

In [11], [14], some practical addressing schemes to achieve or approximate points on the optimum tradeoff curves are given. A comparison of the performance of these methods is available in section VI of this manuscript. The addressing scheme of Lang and Longstaff is further discussed and generalized by Kschischang and Pasupathy in [12] (also refer to [13]).

In comparing different schemes, we need to compute γ_s accurately. Previous methods [11], [12], [15] (also refer to [13]) are based on a continuous approximation. To perform an exact computation, we need the corresponding weight distribution.

II. Weight Distribution of an Optimally Shaped Constellation

The weight distribution of a set of points Λ with respect to a given center is defined as

$$\Theta_\Lambda(q) = \sum_{u \in \Lambda} q^{\|u\|^2} = \sum_{v} C_\Lambda(v) q^{v} \qquad (1)$$

Manuscript received May 13, 1992; revised September 9, 1994. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

A. K. Khandani is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ont. N2L 3G1, Canada.

P. Kabal is with the Department of Electrical Engineering, McGill University, Montreal, P.Q. H3A 2A7, Canada.

IEEE Log Number 9406319.

[1] CER_s is ratio of the number of points used per two dimensions to the minimum necessary number of points per two dimensions [1].

[2] Addressing is the mapping of the data bits to the constellation points.

[3] A third factor is the increase in PAR (peak-to-average-power-ratio), which is uniquely determined by γ_s, CER_s and structure of the 2-D subconstellations [1]. Due to this dependence, we concentrate on the γ_s, CER_s relationship.

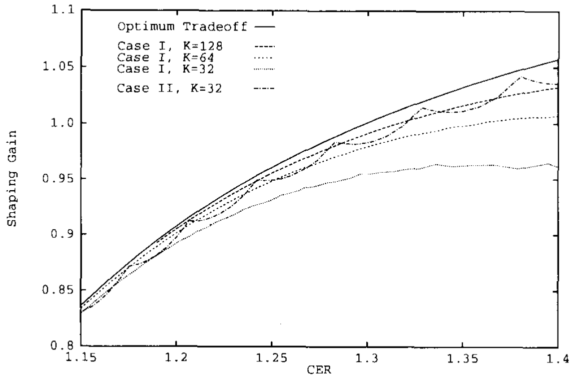

Fig. 1. Trade-off between CER_s and γ_s using K macro-shells in the $N/2$-D subspaces, $N = 32$. Case I corresponds to macro-shells with a fixed number of points and case II corresponds to macro-shells with a fixed number of energy shells.

where $\|u\|^2$ is the norm of the vector associated with point u and $C_\Lambda(v)$ is the number of points of Λ with norm v.

The baseline constellation of cardinality M, denoted as $B_2(M)$, is defined as the set of M points of the least energy from the 2-D half integer grid[‡], $Z^2 + (1/2)^2$. An optimally shaped, N-D constellation is a subset of points of $\{B_2(M)\}^n$, $n = N/2$, of the least energy, where $\{\ \}^n$ denotes the n-fold cartesian product. We have

$$\Theta_{\{B_2(M)\}^n}(q) = [\Theta_{B_2(M)}(q)]^n. \qquad (2)$$

It can be shown that the energy shells of $Z^N + (1/2)^N$ are of values $2i + N/4$, $i = 0, 1, \cdots$, where i is used as the index of the corresponding shell. If B_2 is composed of K energy shells, we obtain $n(K-1)+1$ shells of values $2i + N/4$, $i = 0, \cdots, n(K-1)$ for $\{B_2(M)\}^n$. In $\{B_2(M)\}^n$, unlike $Z^N + (1/2)^N$, shells of indexes $K \leq i \leq n(K-1)$ are partially included and shells with indexes $i > n(K-1)$ are completely discarded.

Define $C_{B_2(M)}(i)$ as the cardinality of the i'th shell of $B_2(M)$. Using 2, we obtain

$$C_{\{B_2(M)\}^n}(i) = \mathrm{DFT}_L^{-1}\{\mathrm{DFT}_L[C_{B_2(M)}(i)]\}^n \qquad (3)$$

where $L = n(K-1)+1$ and DFT_L, DFT_L^{-1} are the L-point discrete Fourier transform and its inverse. Note that $C_{B_2(M)}(i)$ is padded out with zeros.

III. Recursive Merging of Energy Shells

Consider a $B_2(M)$ set composed of K energy shells. In an $N = 2n$-D space, cartesian product of the 2-D shells results in K^n shaping clusters which aggregate into $L = n(K-1)+1 \leq K^n$ shells. A known method to decrease the addressing complexity is based on merging the adjacent 2-D shells into a small number of energy layers (macro-shells) [5]. The merging of shells in [11, [14] is achieved gradually in a hierarchy of stages achieved on the 2-fold cartesian product of the lower dimensional subspaces. In [11], to simplify the addressing, the cardinalities of the macro-shells are restricted to be an integral power of two. In this case, using macro-shells of equal cardinality results in a especially simple scheme. We first explain this approach and then show how one can improve upon it.

Consider an $N = 2^u$-D constellation. We recursively merge energy shells. There are 2^{k_i} macro-shells of equal cardinality in the $N_i = 2^{i+1}$ dimensional subspaces, $i = 0, \cdots, u-2$. In the twofold cartesian product of the N_i-D subspaces, we obtain 2^{2k_i} clusters of equal cardinality. These clusters are arranged in the order of increasing average energy. Then, $2^{k_i - k_{i+1}}$, $i = 0, \cdots, u-3$, subsequent clusters are merged into a higher stage ($2Ni = N_{i+1}$-D) macro-shell. The final constellation is obtained by discarding the N-D clusters with the highest average energy. To achieve the addressing, we need a set of lookup tables to store the components of each macro-shell. The i'th addressing stage, $i = 0, \cdots, u-3$, requires a lookup table with 2^{2k_i} memory locations each with $2k_i$ bits. The last stage requires $2^{2k_{u-2}-r_s}$ memory locations each with $2k_{u-2}$ bits, where $r_s = (N/2)\log_2(\mathrm{CER}_s)$ and CER_s is restricted to have values such that r_s is an integer.

In our experience, for a fixed set of k_i values, $i = 0, \cdots, u-3$, the order in which they are used has almost no effect on the overall performance. Considering that the memory size is a symmetrical function of these values, it is appropriate to select them equal to each other. If they are selected to be unequal (to provide a specific trade-off between complexity and performance), there is a small benefit of using the larger values in the later stages of the hierarchy.

In general, we are looking for efficient, . recursive merging rules that result in macro-shells of integer bit rate. Using macro-shells of equal cardinality (uniform merging), as discussed in [11], is not the best merging rule as is explained in the next section.

IV. Uniform Versus Nonuniform Merging of Clusters

Consider the two-fold cartesian product of a $\{B_2(M)\}^{N/4}$ set. Each of the two $\{B_2\}^{N/4}$ is partitioned into K macro-shells. Consider two merging rules. In case I, macro-shells contain a fixed number of points in the order of increasing energy. In case II, macro-shells contain a fixed number of energy shells. In both cases, in the two-fold cartesian product space, we obtain K^2 clusters. A subset of these clusters of the lowest average energy is selected. Computation of the performance is based on 3. The final result is shown in Fig. 1, which shows the trade-off between CER_s and γ_s. It is seen that using macro-shells with a fixed number of energy shells (case II) results in a better performance. This phenomenon can be justified by considering the hardening effect. Fig. 2 shows the density of points in the energy shells of $\{B_2(256)\}^{N/2}$. It is seen that the points concentrate in a thin energy layer of the space. It should be mentioned that neither of these two merging rules are optimum (in the sense of providing the best trade-off for a given value of K). The performance of a given merging rule also depends on the specific tradeoff point.

Another consideration is the result of the following fact: discarding the clusters of higher energy induces a nonuniform probability distribution on the lower dimensional subspaces such that the clusters of lower energy are used more frequently. This fact is in favor of using a higher resolution in the areas of lower energy. This observation, in conjunction with the hardening effect, suggest decreasing the resolution rather quickly up to regions around the concentration layer and then change it in a slower pace.

In the following, we discuss a practical method for the nonuniform merging of clusters into macro-shells of integer bit rate.

V. Merging of Clusters Using a Binary Tree

Assume that there are 2^k macro-shells of equal cardinality at a given stage of our hierarchy. In the two-fold cartesian product

[‡] The N-D half integer grid, $Z^N + (1/2)^N$, is the collection of the N-D points with components belonging to the set $\{-\infty, \ldots, -3/2, -1/2, 1/2, 3/2, \ldots, +\infty\}$.

TABLE I
PERFORMANCE AND COMPLEXITY OF THE NONUNIFORM MERGING RULE, γ_s dB / MEMORY-SIZE (IN BYTES OF 8 b), FOR $N = 32, (k_0, k_1, k_2, l) = (4, 4, 7, 3)$, NUMBER OF DIFFERENT ℓ'S AND $S = 1$

$\ell=7$	6	5	4	3	2	1	CER$_s$ = 1.1	CER$_s$ = 1.2	CER$_s$ = 1.3	CER$_s$ = 1.4
2	1	1	1	1	1	1	0.67 dB/1.63 k	0.84 dB/1.47 k	0.93 dB/1.30 k	0.98 dB/1.14 k
·	2	1	1	1	3	·	0.70 dB/1.53 k	0.86 dB/1.27 k	0.94 dB/1.06 k	0.98 dB/0.89 k
·	2	1	1	3	·	1	0.67 dB/1.47 k	0.84 dB/1.26 k	0.93 dB/1.07 k	0.98 dB/0.90 k
·	2	1	3	·	1	1	0.67 dB/1.63 k	0.84 dB/1.45 k	0.93 dB/1.24 k	0.98 dB/1.04 k
·	2	3	·	1	1	1	0.67 dB/1.42 k	0.84 dB/1.26 k	0.92 dB/1.09 k	0.98 dB/0.94 k
·	4	·	1	1	1	1	0.67 dB/1.42 k	0.84 dB/1.26 k	0.93 dB/1.08 k	0.97 dB/0.92 k
·	·	2	1	3	2	·	0.70 dB/1.38 k	0.87 dB/1.07 k	0.94 dB/0.86 k	0.98 dB/0.73 k
·	·	2	3	·	3	·	0.70 dB/1.30 k	0.86 dB/1.02 k	0.94 dB/0.83 k	0.97 dB/0.71 k
·	·	2	3	2	·	1	0.67 dB/1.14 k	0.84 dB/0.96 k	0.93 dB/0.81 k	0.97 dB/0.70 k
·	·	4	·	1	3	·	0.70 dB/1.03 k	0.86 dB/0.85 k	0.93 dB/0.73 k	0.96 dB/0.65 k
·	·	4	·	3	·	1	0.67 dB/1.17 k	0.83 dB/1.04 k	0.91 dB/0.88 k	0.94 dB/0.75 k
·	·	4	2	·	1	1	0.67 dB/1.17 k	0.83 dB/1.04 k	0.91 dB/0.88 k	0.94 dB/0.75 k
·	·	6	·	·	1	1	0.67 dB/0.89 k	0.83 dB/0.82 k	0.91 dB/0.70 k	0.95 dB/0.63 k
·	·	·	2	5	1	·	0.70 dB/0.83 k	0.85 dB/0.69 k	0.91 dB/0.62 k	0.92 dB/0.58 k
·	·	·	4	2	2	·	0.70 dB/0.86 k	0.85 dB/0.68 k	0.90 dB/0.59 k	0.88 dB/0.56 k
·	·	·	6	1	·	1	0.66 dB/0.90 k	0.82 dB/0.67 k	0.87 dB/0.60 k	0.88 dB/0.57 k
·	·	·	·	8	·	·	—	—	—	—

TABLE II
PERFORMANCE AND COMPLEXITY OF THE NONUNIFORM MERGING RULE, γ_s dB / MEMORY-SIZE (IN BYTES OF 8 b), FOR $N = 32, (k_0, k_1, k_2, l) = (4, 4, 7, 3)$, NUMBER OF DIFFERENT ℓ'S AND $S = 2$

$\ell=7$	6	5	4	3	2	1	CER$_s$ = 1.1	CER$_s$ = 1.2	CER$_s$ = 1.3	CER$_s$ = 1.4
2	1	1	1	1	1	1	0.70 dB/1.38 k	0.85 dB/1.25 k	0.93 dB/1.10 k	0.98 dB/0.99 k
·	2	1	1	1	3	·	0.71 dB/1.42 k	0.86 dB/1.18 k	0.94 dB/0.99 k	0.99 dB/0.84 k
·	2	1	1	3	·	1	0.70 dB/1.13 k	0.86 dB/1.02 k	0.94 dB/0.90 k	0.98 dB/0.80 k
·	2	1	3	·	1	1	0.71 dB/1.20 k	0.86 dB/1.08 k	0.94 dB/0.95 k	0.98 dB/0.84 k
·	2	3	·	1	1	1	0.68 dB/1.15 k	0.84 dB/1.05 k	0.93 dB/0.93 k	0.97 dB/0.84 k
·	4	·	1	1	1	1	0.68 dB/1.09 k	0.84 dB/0.99 k	0.93 dB/0.86 k	0.97 dB/0.75 k
·	·	2	1	3	2	·	0.71 dB/1.08 k	0.88 dB/0.88 k	0.94 dB/0.75 k	0.97 dB/0.67 k
·	·	2	3	·	3	·	0.71 dB/1.18 k	0.86 dB/0.94 k	0.94 dB/0.78 k	0.97 dB/0.68 k
·	·	2	3	2	·	1	0.70 dB/0.98 k	0.85 dB/0.86 k	0.93 dB/0.75 k	0.97 dB/0.68 k
·	·	4	·	1	3	·	0.71 dB/0.80 k	0.85 dB/0.70 k	0.91 dB/0.63 k	0.94 dB/0.61 k
·	·	4	·	3	·	1	0.69 dB/1.04 k	0.84 dB/0.94 k	0.91 dB/0.81 k	0.94 dB/0.72 k
·	·	4	2	·	1	1	0.69 dB/1.04 k	0.84 dB/0.94 k	0.91 dB/0.81 k	0.94 dB/0.72 k
·	·	6	·	·	1	1	0.73 dB/0.77 k	0.86 dB/0.70 k	0.92 dB/0.64 k	0.94 dB/0.60 k
·	·	·	2	5	1	·	0.70 dB/0.79 k	0.85 dB/0.67 k	0.91 dB/0.62 k	0.92 dB/0.58 k
·	·	·	4	2	2	·	0.70 dB/0.82 k	0.85 dB/0.67 k	0.90 dB/0.59 k	0.88 dB/0.56 k
·	·	·	6	1	·	1	0.67 dB/0.86 k	0.82 dB/0.66 k	0.87 dB/0.59 k	0.88 dB/0.57 k
·	·	·	·	8	·	·	—	—	—	—

TABLE III
PERFORMANCE AND COMPLEXITY OF THE NONUNIFORM MERGING RULE, γ_s dB / MEMORY-SIZE (IN BYTES OF 8 b), FOR $N = 32, (k_0, k_1, k_2, l) = (4, 4, 7, 3)$, NUMBER OF DIFFERENT ℓ'S AND $S = 3$

$\ell=7$	6	5	4	3	2	1	CER$_s$ = 1.1	CER$_s$ = 1.2	CER$_s$ = 1.3	CER$_s$ = 1.4
2	1	1	1	1	1	1	0.73 dB/1.20 k	0.86 dB/1.09 k	0.93 dB/0.98 k	0.97 dB/0.88 k
·	2	1	1	1	3	·	0.70 dB/1.34 k	0.86 dB/1.11 k	0.94 dB/0.94 k	0.98 dB/0.81 k
·	2	1	1	3	·	1	0.66 dB/0.99 k	0.83 dB/0.89 k	0.93 dB/0.81 k	0.98 dB/0.74 k
·	2	1	3	·	1	1	0.73 dB/1.04 k	0.86 dB/0.94 k	0.93 dB/0.86 k	0.97 dB/0.78 k
·	2	3	·	1	1	1	0.67 dB/1.08 k	0.83 dB/1.00 k	0.92 dB/0.92 k	0.97 dB/0.84 k
·	4	·	1	1	1	1	0.70 dB/0.99 k	0.84 dB/0.91 k	0.92 dB/0.82 k	0.96 dB/0.73 k
·	·	2	1	3	2	·	0.70 dB/1.00 k	0.86 dB/0.83 k	0.95 dB/0.72 k	0.97 dB/0.65 k
·	·	2	3	·	3	·	0.71 dB/1.18 k	0.86 dB/0.94 k	0.94 dB/0.78 k	0.97 dB/0.68 k
·	·	2	3	2	·	1	0.68 dB/0.91 k	0.83 dB/0.81 k	0.93 dB/0.72 k	0.97 dB/0.66 k
·	·	4	·	1	3	·	0.71 dB/0.80 k	0.85 dB/0.70 k	0.91 dB/0.63 k	0.94 dB/0.61 k
·	·	4	·	3	·	1	0.67 dB/0.98 k	0.82 dB/0.91 k	0.90 dB/0.82 k	0.94 dB/0.72 k
·	·	4	2	·	1	1	0.67 dB/0.98 k	0.82 dB/0.91 k	0.90 dB/0.82 k	0.94 dB/0.72 k
·	·	6	·	·	1	1	0.73 dB/0.77 k	0.86 dB/0.70 k	0.92 dB/0.64 k	0.94 dB/0.60 k
·	·	·	2	5	1	·	0.70 dB/0.79 k	0.85 dB/0.67 k	0.91 dB/0.62 k	0.92 dB/0.58 k
·	·	·	4	2	2	·	0.70 dB/0.78 k	0.85 dB/0.66 k	0.90 dB/0.59 k	0.88 dB/0.56 k
·	·	·	6	1	·	1	0.67 dB/0.86 k	0.82 dB/0.66 k	0.87 dB/0.59 k	0.88 dB/0.57 k
·	·	·	·	8	·	·	—	—	—	—

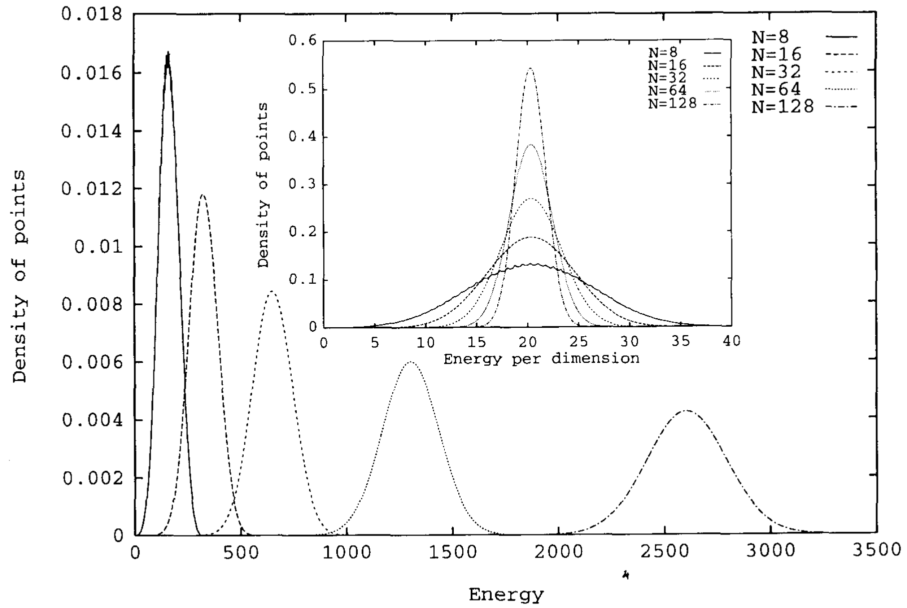

Fig. 2. Density of points of $\{B_2(256)\}^{N/2}$ as a function of energy or energy per dimension, $N = 8, 16, 32, 64, 128$.

TABLE IV
PERFORMANCE AND COMPLEXITY OF THE NONUNIFORM MERGING RULE, $N = 32$, $(k_0, k_1, k_2, l) = (4, 4, 7, 3)$. THE OPTIMUM VALUES OF γ_s ARE WRITTEN IN PARENTHESES

CER_s	γ_s (dB)/Memory (Byte)
1.1	0.73 (0.73) dB / 0.77 k
1.2	0.88 (0.91) dB / 0.88 k
1.3	0.95 (1.00) dB / 0.72 k
1.4	0.99 (1.05) dB / 0.84 k

TABLE V
PERFORMANCE AND COMPLEXITY OF THE UNIFORM MERGING RULE, (METHOD OF [11]), γ_s dB/MEMORY-SIZE (IN BYTES OF 8 BITS), FOR $N = 32$ AND DIFFERENT VALUES OF (k_0, k_1, k_2, k_3).

k_0	k_1	k_2	k_3	$\text{CER}_s = 1.1$	$\text{CER}_s = 1.2$	$\text{CER}_s = 1.3$	$\text{CER}_s = 1.4$
4	4	4	7	0.72 dB/6.84 k	0.86 dB/2.26 k	0.91 dB/1.17 k	0.87 dB/0.88 k
4	4	4	8	0.72 dB/28.6 k	0.87 dB/7.67 k	0.93 dB/2.67 k	0.94 dB/1.34 k
4	4	5	8	0.72 dB/29.6 k	0.88 dB/8.67 k	0.95 dB/3.67 k	0.97 dB/2.34 k
4	4	5	9	0.72 dB/127 k	0.88 dB/32.9 k	0.96 dB/10.4 k	0.99 dB/4.39 k
4	5	5	8	0.73 dB/30.6 k	0.89 dB/9.67 k	0.96 dB/4.67 k	0.98 dB/3.34 k
4	5	5	9	0.72 dB/128 k	0.88 dB/33.9 k	0.96 dB/11.4 k	0.99 dB/5.39 k
4	5	6	7	0.72 dB/13.6 k	0.89 dB/9.00 k	0.95 dB/7.92 k	0.94 dB/7.63 k
4	5	6	8	0.73 dB/35.3 k	0.89 dB/14.4 k	0.97 dB/9.42 k	1.00 dB/8.09 k
4	5	6	9	0.73 dB/133 k	0.90 dB/38.7 k	0.98 dB/16.1 k	1.02 dB/10.1 k
5	5	5	8	0.73 dB/31.6 k	0.89 dB/10.7 k	0.97 dB/5.67 k	1.00 dB/4.34 k
5	5	5	9	0.73 dB/129 k	0.89 dB/34.9 k	0.97 dB/12.4 k	1.00 dB/6.40 k
5	5	6	8	0.73 dB/36.4 k	0.90 dB/15.4 k	0.98 dB/10.4 k	1.01 dB/9.09 k
5	5	6	9	0.73 dB/134 k	0.90 dB/39.6 k	0.98 dB/17.2 k	1.02 dB/11.1 k

space, we obtain 2^{2k} clusters that are merged into 2^l macro-shells of integer bit rate. Define $2^{-\ell_i}$ to be the fraction of the number of clusters in the ith macro-shell, $i = 0, \cdots, 2^l - 1$. The ℓ_i's satisfying $\Sigma_i 2^{-\ell_i} = 1$. A simple argument shows that the ℓ_i's can be selected as the lengths of different paths in any binary tree with $2^l - 1$ intermediate nodes (resulting in 2^l final nodes). As the number of such trees is usually quite small, one can use an exhaustive search to find the best tree for a specific trade-off between CER_s and γ_s. This configuration allows to use a set of prefix codes for the addressing of the macro-shells. The idea of using a prefix coding scheme for the addressing is also discussed in a different context in [11]. The approach presented here is much more efficient.

This nonuniform merging rule is applied in the $(u - 2)$th stage (stage indexed by $u - 3$) of the hierarchy. The corresponding merging rule for the $(u - 1)$th stage is as follows: if there are an integral power of two of successive macro-shells with equal cardinality, these are merged into a single, larger macro-shell. One can also apply this rule successively several times. The number of successive times is denoted by S. The performance and complexity of this approach is shown in Tables I–III. These tables correspond to $S = 1, 2, 3$, and each table contains all the possible combinations of ℓ_i's, $i = 0, \cdots, 7$. For example, the first row in each table means that: $(\ell_i, i = 0, \cdots, 7) = (7, 7, 6, 5, 4, 3, 2, 1)$ and the second row means that: $(\ell_i, i = 0, \cdots, 7) = (6, 6, 5, 4, 3, 2, 2, 2)$. The cases of special interest (good performance and low complexity) are underlined.

We have also examined: i) the case of $S = 0$, and ii) applying the nonuniform merging in the $(u - 1)$'th stage. In both cases the results were inferior to those presented here.

VI. Numerical Comparisons

A four state trellis diagram of [6] achieves $\gamma_s = 0.95$ dB, $\mathrm{CER}_s = 1.5$. In [14], an example for $N = 64$ is given which needs 1440 multiply-adds (assuming a 16 bit processor) and a memory of 1.5 kilo-bytes to achieve a tradeoff point with $\gamma_s = 1.15$ dB, $\mathrm{CER}_2 = 1.5$

For a given CER_s by appropriately choosing the merging parameters, we achieve nearly all of the shaping gain possible using a small amount of memory (refer to Table IV). Computation of the optimum γ_s is based on 3.

Table IV can be compared to Table V, which shows the method applied when an equal number of points is used in the macro-shells at each stage (this becomes the method discussed in [11]). The cases of special interest are underlined. The present schemes offer a reduction in complexity by a factor of 5 to 10.

VII. Summary and Conclusions

We have presented efficient addressing schemes based on partitioning the subconstellations into nonuniform shaping macro-shells of integer bit rate. The corresponding shaping performance is computed using the weight distribution of an optimally shaped constellation. As an example of performance in a 32-D space, we use about 0.8 k-bytes of memory to achieve trade-off points very close to the optimum performance. It seems that this is the simplest known method to achieve shaping gains in the order of 1.0 dB. Note that this method needs only a small number of table lookups and no arithmetic operation is needed.

References

[1] G. D. Forney, Jr. and L. F. Wei, "Multidimensional constellations—Part I: Introduction, figures of merit, and generalized cross constellations," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 877–892, Aug. 1989.

[2] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 106–109, Jan. 1985.

[3] L. F. Wei, "Trellis coded modulation with multidimensional constellations," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 483–501, July 1987.

[4] G. D. Forney, Jr., "Multidimensional constellations—Part II: Voronoi constellations," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 941–958, Aug. 1989.

[5] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Trans. Inform. Theory*, vol. 36, pp. 726–740, July 1990.

[6] G. D. Forney, Jr., "Trellis shaping," *IEEE Trans. Inform. Theory*, vol. 38, pp. 281–300, Mar. 1992.

[7] G. R. Lang and F. M. Longstaff, "A leech lattice modem," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 968–973, Aug. 1989.

[8] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Trans. Inform. Theory*, vol. 39, May 1993.

[9] J. R. Livingston, "Shaping using variable-size regions," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1347–1353, July 1992.

[10] A. R. Calderbank and M. Klimesh, "Balanced codes and nonequiprobable signaling," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 1119–1122, May 1992.

[11] A. K. Khandani and P. Kabal, "Shaping multi-dimensional signal spaces—Part II: Shell-addressed constellations," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1809–1819, Nov. 1993.

[12] F. R. Kschischang and S. Pasupathy, "Optimal shaping properties of the truncated polydisc," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 892–903, May 1994.

[13] F. R. Kschischang, "Shaping and coding gain criteria in signal constellation design," Ph.D. dissertation, Toronto Univ., Toronto, Ont., Canada, June 1991.

[14] R. Laroia, N. Farvardin, and S. A. Tretter "On optimal shaping of multi-dimensional constellations," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 1044–1056, July 1994.

[15] A. K. Khandani and P. Kabal, "Shaping multi-dimensional signal spaces—Part I: Optimum shaping, shell mapping," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1799–1808, Nov. 1993.

0018-9448/94$04.00 © 1994 IEEE